AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior version, and listings, of claims in the application.

Listing of Claims:

1 (currently amended): A system for reducing latency of computer operations, comprising:

a first processing pipeline comprising a prevalidated cache translation lookaside buffer (TLB), the prevalidated cache TLB comprising a virtual address (VA) content addressable memory (CAM), wherein the VA CAM receives virtual address information for integer load operations; and

a second processing pipeline, independent of the first processing pipeline, the second processing pipeline comprising:

a cache tag array that holds physical addresses of cache lines,

a master TLB that receives virtual address information for store operations and generates a physical address,

a bypass around the master TLB, wherein if ~~the~~ a store address is a physical address, the physical address bypasses the master TLB, and

a comparator that compares a physical address from one of the bypass and the master TLB to a physical address from the cache tag array, wherein if the physical addresses match, a store/invalidate cache way hit is generated.

2. (original): The system of claim 1, wherein the first processing pipeline further comprises:

a data cache;

prevalidated cache tag array; and

a logic circuit at the output of the VA CAM and the prevalidated cache tag array to produce a cache way hit.

3. (original): The system of claim 2, wherein the first processing pipeline further comprises a multiplexor that uses the cache way hit to select a data cache output.

4. (currently amended): The system of claim 1, wherein the first processing pipeline further comprises a column clear module that clears TLB hit bits of ~~the~~ a prevalidated cache tag array for one or more TLB slot entry positions.

5. (original): The system of claim 1, wherein the prevalidated cache TLB sets TLB hit bits for all TLB slots to be invalidated.

6. (currently amended): The system of claim 1, further comprising a hardware cleansing function that removes stale cache lines after each prevalidated cache TLB insert or purge, wherein the function:

compares each cache tag index that has an invalid entry in ~~the~~ a prevalidated cache ~~tags~~ tag array with a valid entry in ~~the~~ a physical address cache ~~tags~~ tag array; and

clears each physical address cache tag when the comparison results in a match.

DT 2/4/05

7 (currently amended ~~original~~): The system of claim 1, further comprising:

DT 2/4/05

an apparatus to invalidate store valid bits whenever a TLB bit in ~~the~~ a prevalidated cache tag array is cleared due to a column clear operation, wherein the apparatus.

detects a column clear function;

ORs the multiple bits; and

compares each bit invalidated in the prevalidated cache tag array to each index row of the store valid bits in both the prevalidated cache tag array and the physical address cache tag. array

DT 2/4/05

8. (currently amended): The system of claim 1, further comprising:

a stale cache line control to invalidate store valid bits on one index location when an index is being loaded with a new cache line, wherein the stale cache line control:

receives cache way hit information and, optionally, column clear information;

provides signals to a store valid bit module located in both the first processing pipeline and the second processing pipeline,

sends the address of a cache line that is being processed in a cache line fill through the physical address cache tag;

determines if the cache line is stale in any of the cache ways that are not being loaded with the new cache line whenever there is a hit on any of the other cache ways;

invalidates the stale cache line; and

clears the store valid bits in both store valid bit modules.

9. (currently amended): A method for reducing latency of computer operations, comprising:

running a first processing pipeline wherein the first processing pipeline receives virtual address information for integer load operations from a pre-validated cache TLB with a VA CAM; and

running a second processing pipeline, independent of the first processing pipeline, wherein the second processing pipeline:

holds physical addresses of cache lines in a physical address cache tag array.

receives address information for store operations in a master TLB, wherein the master TLB generates a physical address;

bypasses around the master TLB if the store address information is a physical address;

compares the physical address from one of the bypass and the master TLB to a physical address from the physical cache tag array; and

DT 2/4/05

generates a cache way hit if the physical addresses match.

10. (currently amended): The method of claim 9, further comprising:

clearing entries from ~~the~~ a prevalidated cache tag array for one or more prevalidated cache TLB slot entry positions.

DT 2/4/05

11. Currently amended (~~original~~): The method of claim 9, further comprising:

invalidating prevalidated cache tags by:

setting prevalidated cache TLB hit bits for all prevalidated cache TLB slots that are to be invalidated;

DT 2/4/05

receiving a column clear signal by ~~the~~ a prevalidated cache tag array; and

clearing all prevalidated cache TLB hit bits in the prevalidated cache tag array for all TLB slots indicated by the TLB hit signals.

12. (currently amended): The method of claim 9, further comprising:

removing cache lines that are invalidated by a column clear but remain valid for a stale cache line by:

comparing each cache tag index that has an invalid entry in ~~the~~ a prevalidated cache tag array with a valid entry in the physical address cache tag array, and

clearing each physical address cache tag when the comparison results in a match.

13. (currently amended): The method of claim 9, further comprising:

detecting a column clear function;

connecting the column clear function information to each index row of ~~the~~ store valid bits in both ~~the~~ a prevalidated cache tag array and the physical address cache tag array.

DT 2/4/05

14. (original): The method of claim 13, further comprising OR-ing multiple store valid bits.

15 (currently amended). The method of claim 9, further comprising:

removing stale cache lines by.

receiving cache way hit information and optionally column clear information;

providing signals to a store valid bit module located within the first processing pipeline and the second processing pipeline;

sending ~~the~~ an address of a cache line that is being processed ~~in the cache~~ through the physical address cache tag array;

determining if the cache line is stale in any of the cache ways that are not being loaded with the new cache line whenever there is a hit on any of the other cache ways; and

DT 2/4/05

invalidating the stale cache line and clearing the store valid bits in the store valid bit ~~module~~ modules located in the first processing pipeline and the second processing pipeline.